



MUGSy Demo

**Paul Capotosto
Yuri Frankel
Maria Jacob
Gary Orr
June Phillips
Steve Robinson
Darlene West**



What is MUGSy??

- ❖ An integrated environment that provides:
 - Engineering Repository Management
 - Configuration Management
 - Change Management
 - Test Management



Presentation Goals

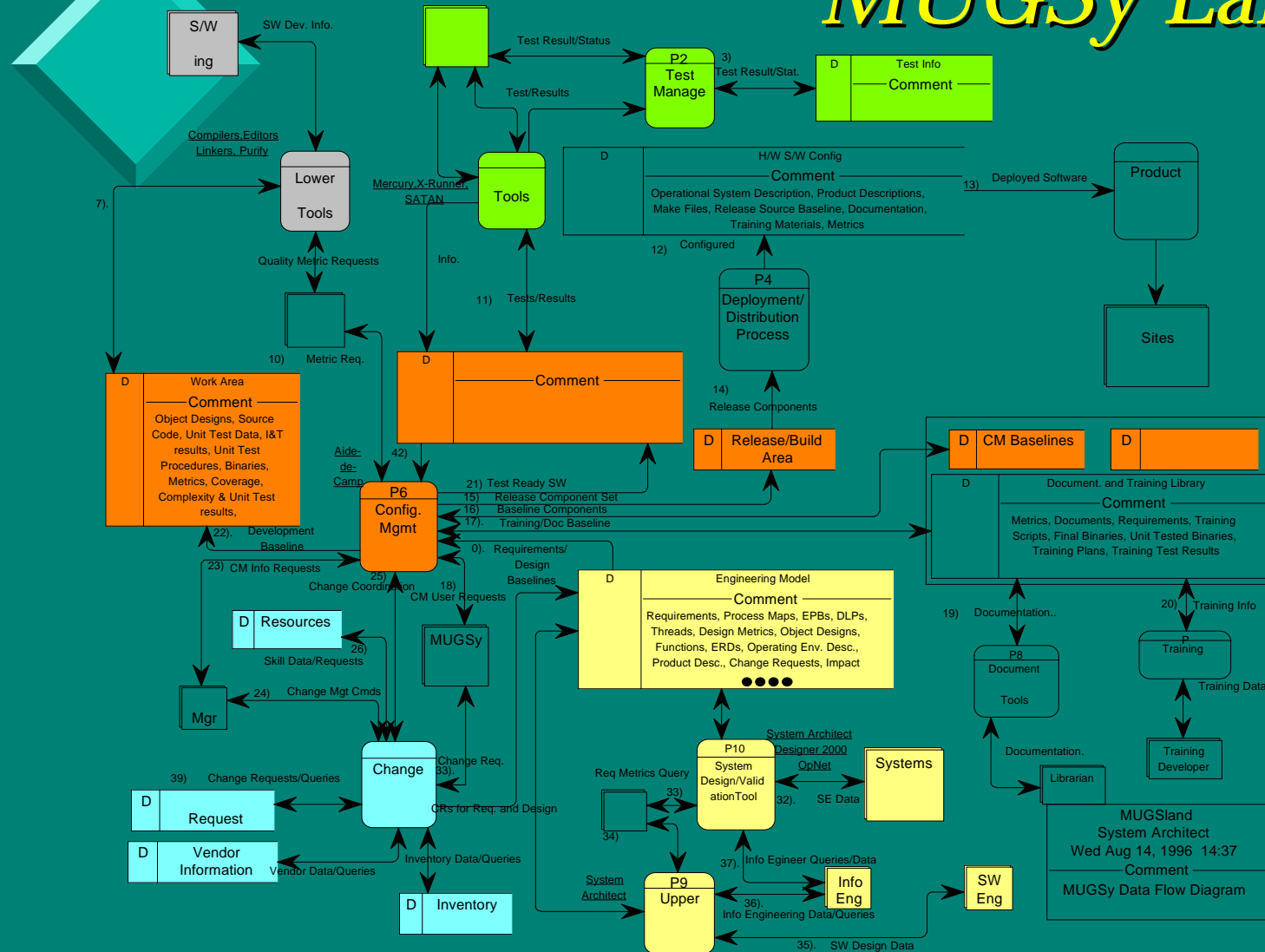
*To describe the
end-to-end change request process and
demonstrate MUGSy as an
integrated environment supporting this
process*

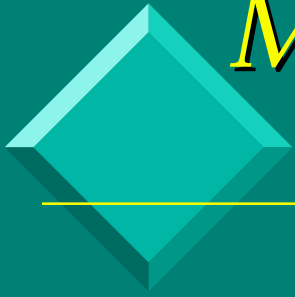


Agenda

- ❖ Change Request Process Flow and Demonstration
- ❖ Configuration Management
- ❖ Engineering Repository

MUGSy Land





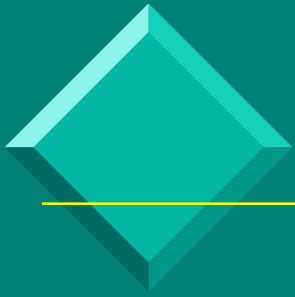
MUGSy Capabilities for the Design Environment

- ❖ Provided using the Popkin System Architect (SA) that has been used to capture:
 - Requirements
 - TDA for Release 1 and 2
 - CI design for Release 1 and 2
 - Test threads



MUGSy Capabilities for Configuration Management

- ❖ Provided using the True Software Aide-de-Camp (ADC) /Pro. The ADC repositories and directories have been established for all CIs and ADC is being used to baseline Design walkthrough and Code walkthrough packages



MUGSy Capabilities for Change Management

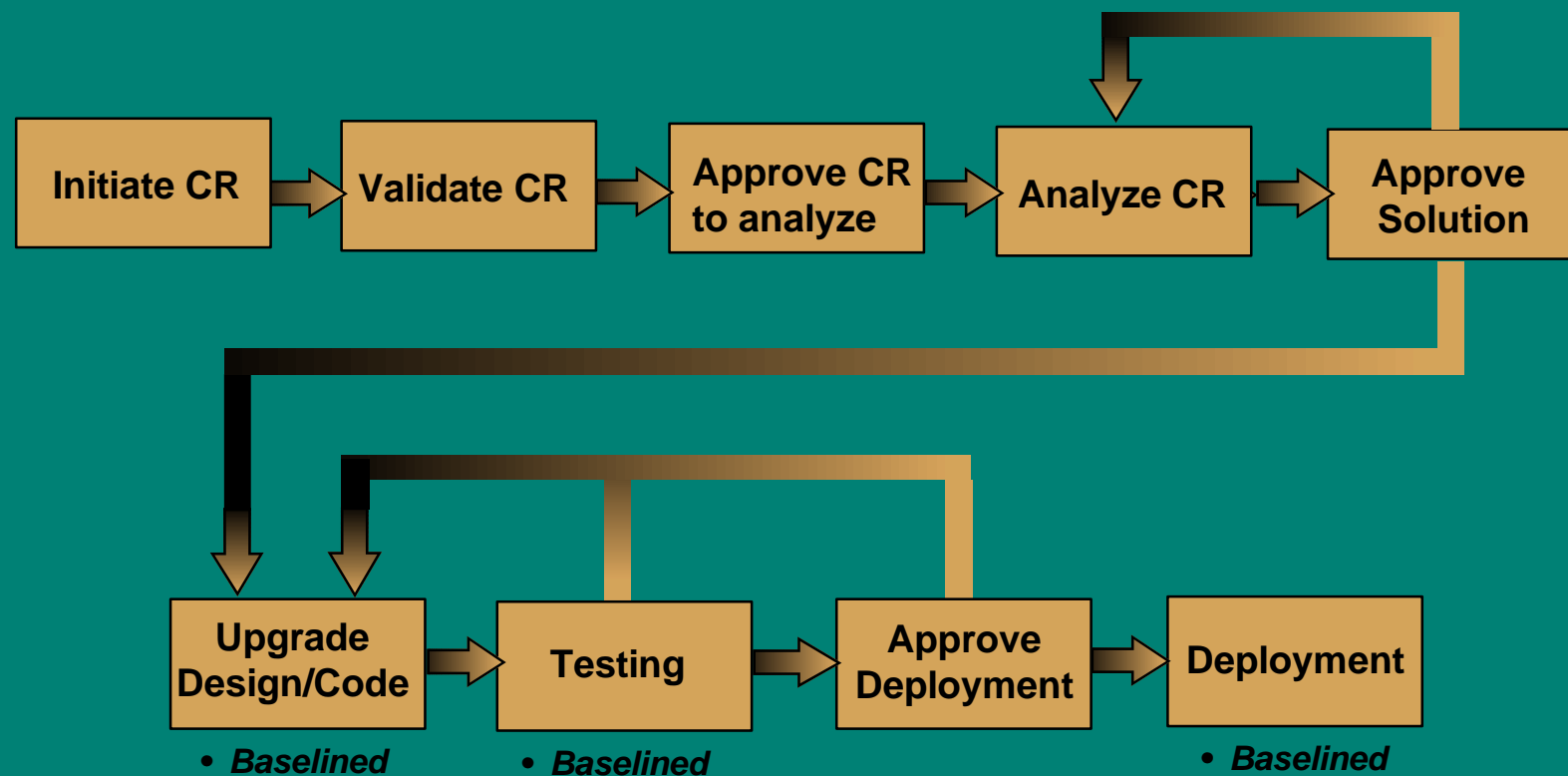
- ❖ A single user interface, called MUGSy Central, based on SCOPUS, has been developed that is being used to:
 - baseline CCS products
 - update the CCS roster
 - identify, track and provide automatic notification of Change Requests (CRs) against the Office Environment or any CCS products



Assumptions for Change Request Process

- ❖ Approval at appropriate level:
 - CCS Functional Team (CFT)
 - Release Manager (RM)
 - CCS Architecture Board (CAB)
 - System Architecture Board (SAB)
- ❖ Every CCS Product has a pre-defined owner and approver
- ❖ Change request process described and demonstrated assumes CR submitted against operational baseline

Change Process: At a high level

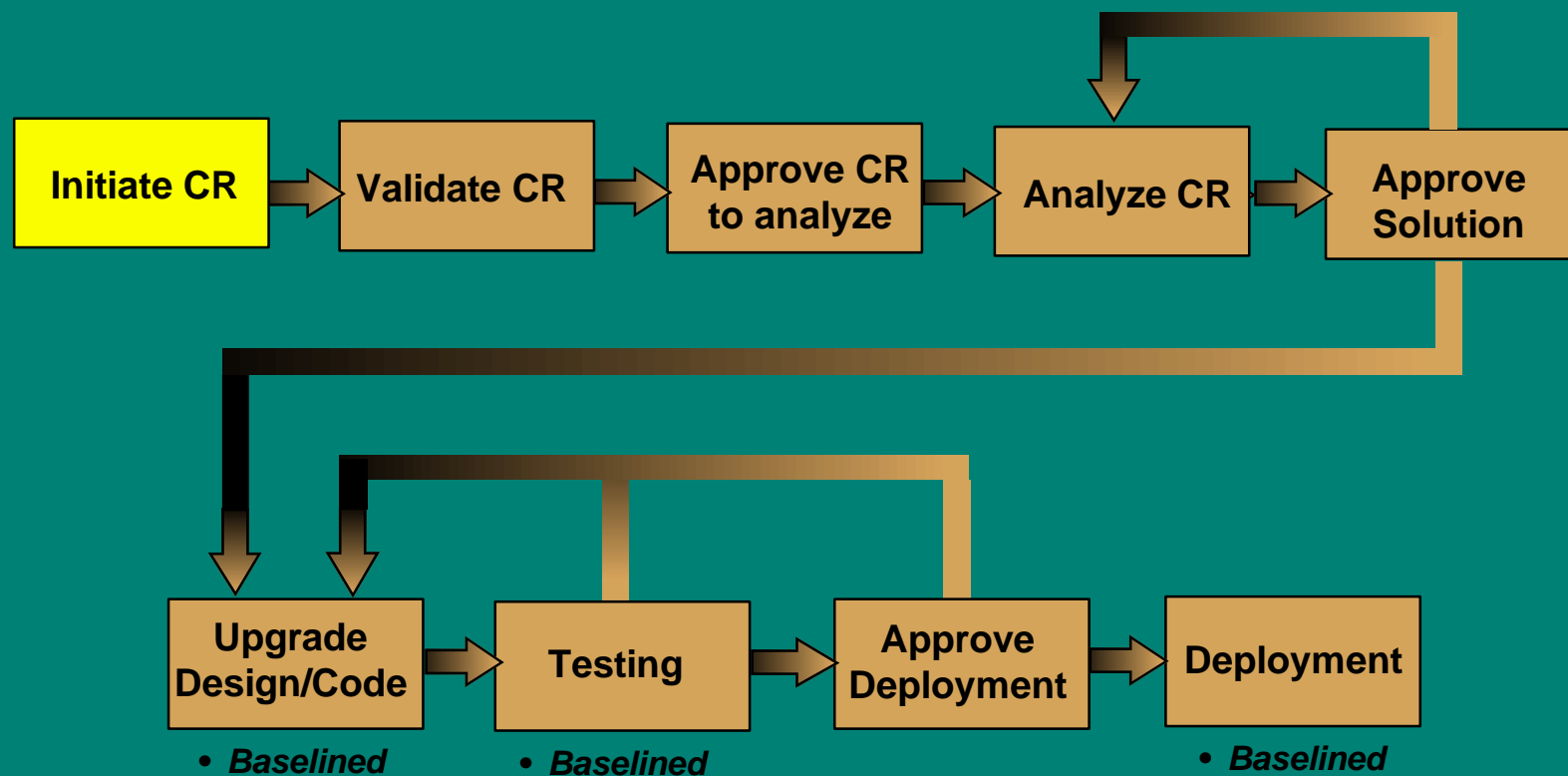




Roles during the Change Request Process

- ❖ Originator: Any MUGSy user entering a CR.
- ❖ Product Owner: person responsible for creating/maintaining a specific CCS product
- ❖ CR Assignee: person responsible for ensuring the CR process is completed. May actually “do some of the work”
- ❖ Customer: accepts and installs operational CCS
- ❖ Review Team: team designated to review and approve CR at specific steps

Initiate CR

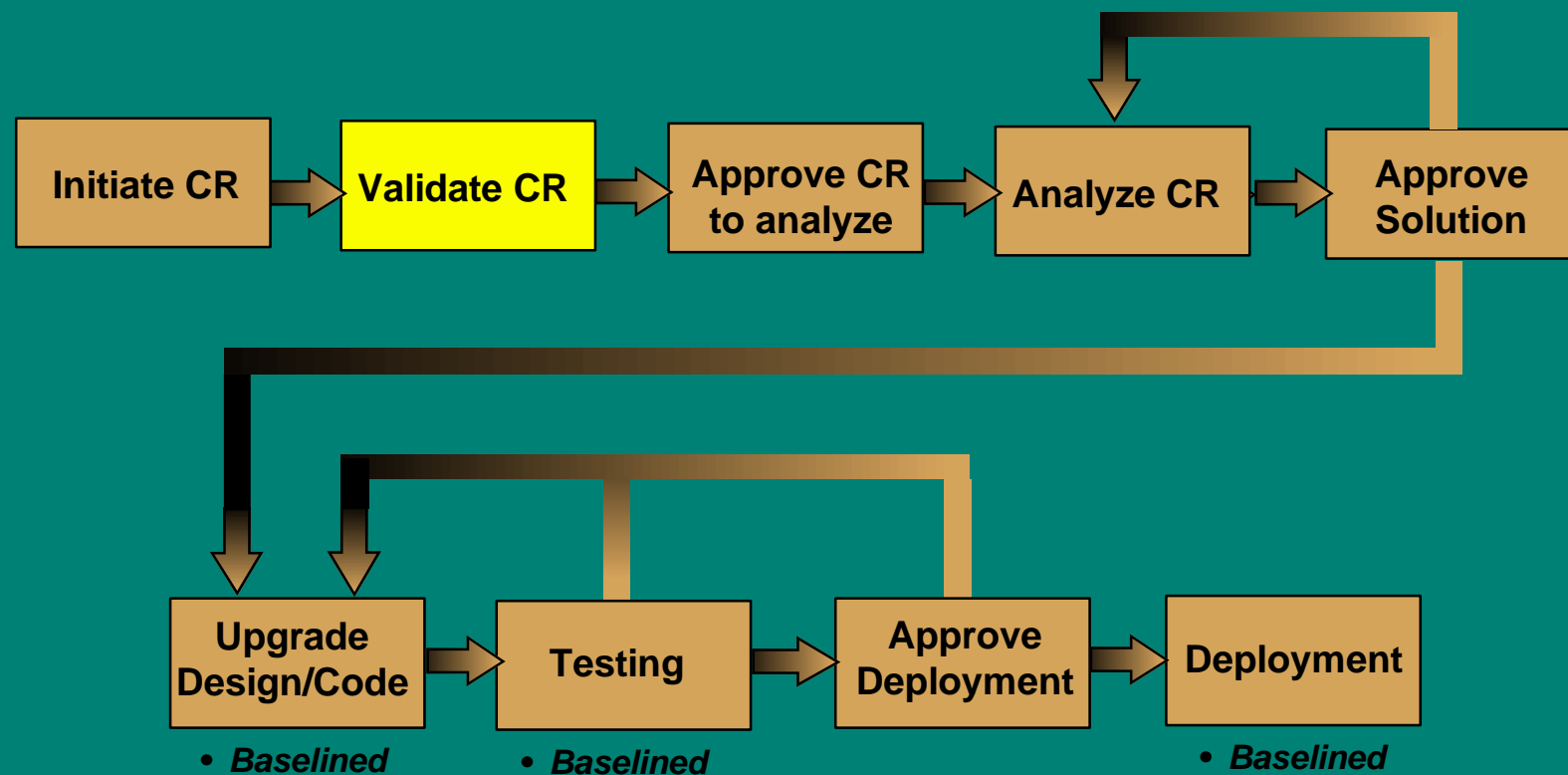




Initiate CR

USER/TOOL	DESCRIPTION
MUGSy User (Originator)	1 -- Enters CR
Scopus/Oracle	2 -- Captures the “ <i>Reported</i> ” CR 3 -- Assigns CR to product owner 4 -- Notifies Review Team & product owner

Validate CR

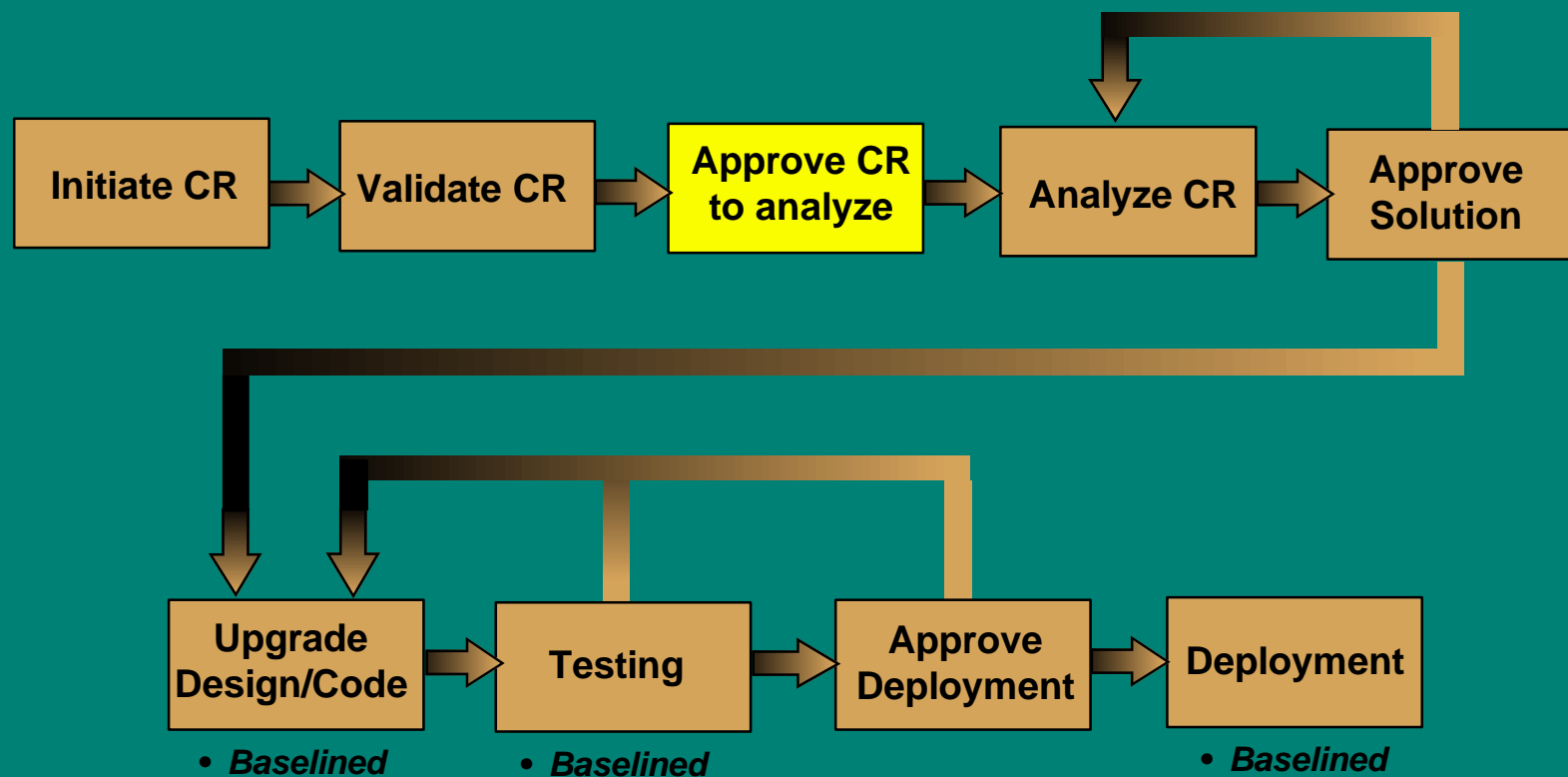




Validate CR

USER/TOOL	DESCRIPTION
Product Owner	5 -- Reviews/Validates CR ; assigns class 6 -- Updates the CR Status to “Open”
Scopus/Oracle	7 -- Captures status of CR 8 -- Notifies Originator & Review Team

Approve CR

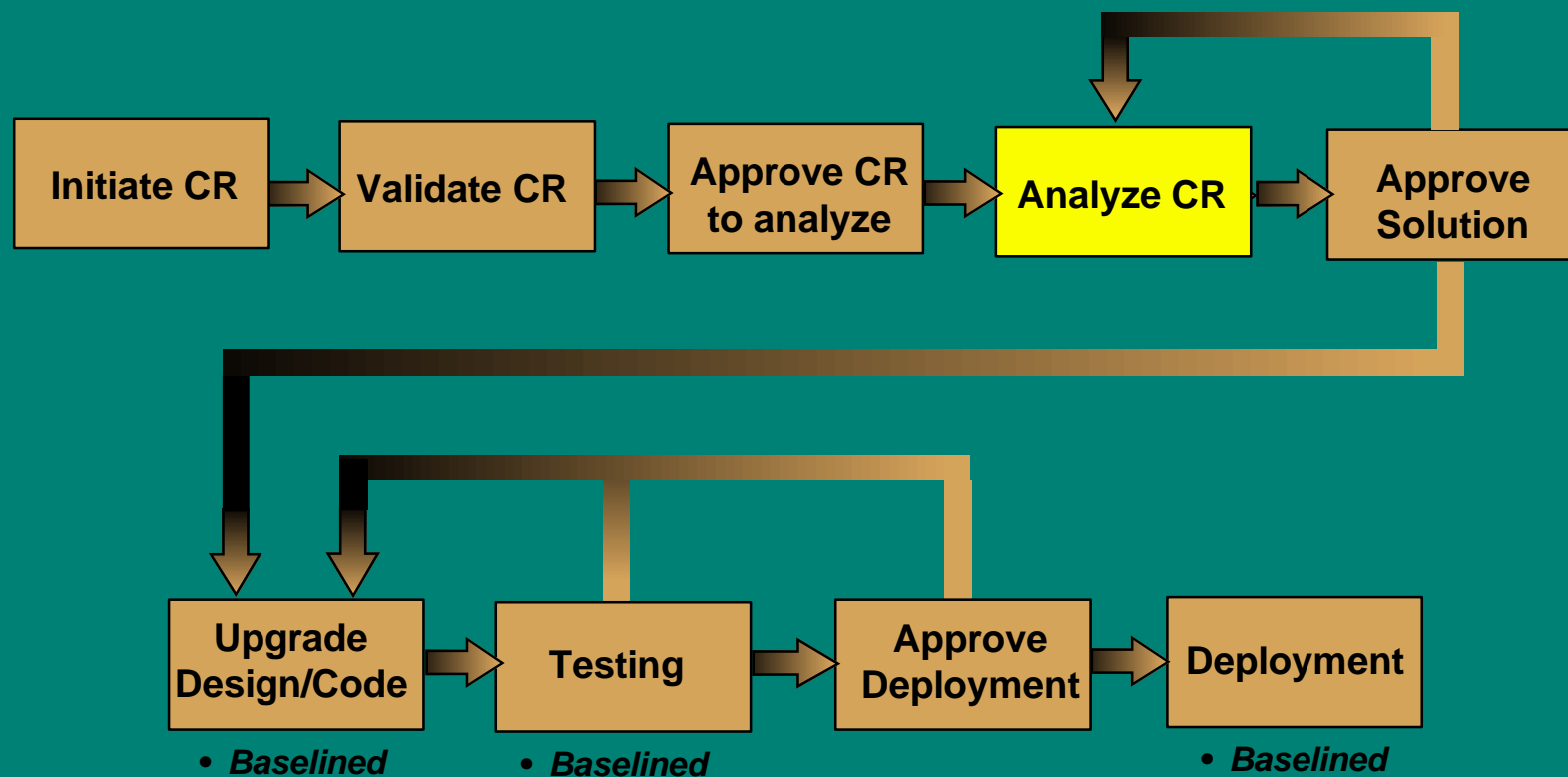




Approve CR to analyze

USER/TOOL	DESCRIPTION
Review Team (CFT, RM, CAB or SAB)	9 -- Approves CR Class and assigns 10 -- Updates CR status to <i>“Assigned”</i>
Scopus/Oracle	11 -- Captures CR status & assignee 12 -- Notifies Review Team, Originator & Assignee

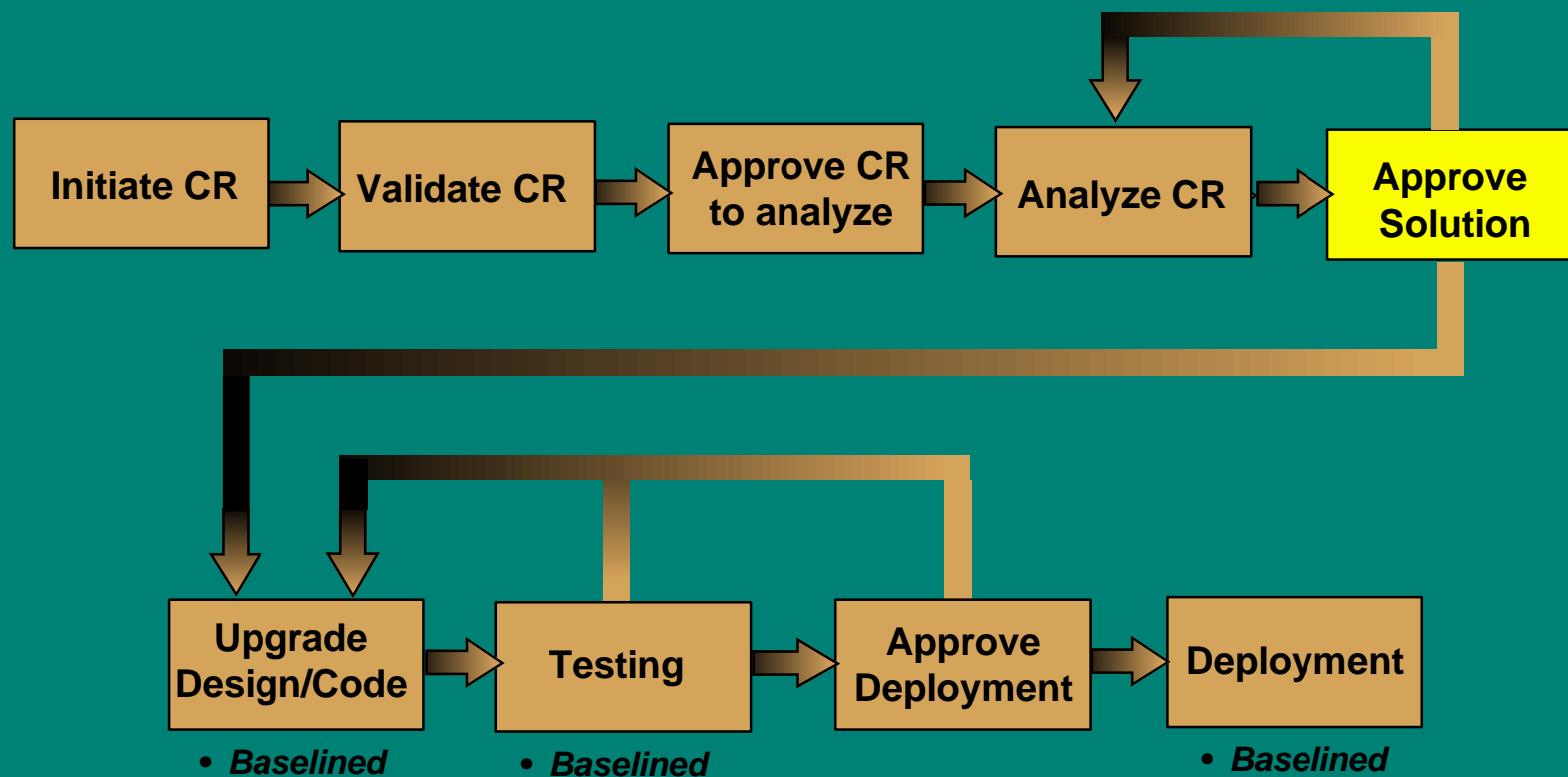
Analyze CR



Analyze CR

USER/TOOL	DESCRIPTION
Aide-de-Camp	13 -- Supplies current baseline (Read Only)
System Architect	14 -- Provides current baselined requirements & design
CR Assignee	15 -- Identifies dependencies/associate CRs using baselined information, updates CR 16 -- Provides solutions/additional info
MUGSy Users	17 -- Provides solutions/additional info
CR Assignee	18 -- Consolidates solutions and recommends a solution 19 -- Updates CR status to <i>"Analysis Complete"</i>
Scopus/Oracle	20 -- Captures new CR information 21 -- Notifies Originator & Review Team

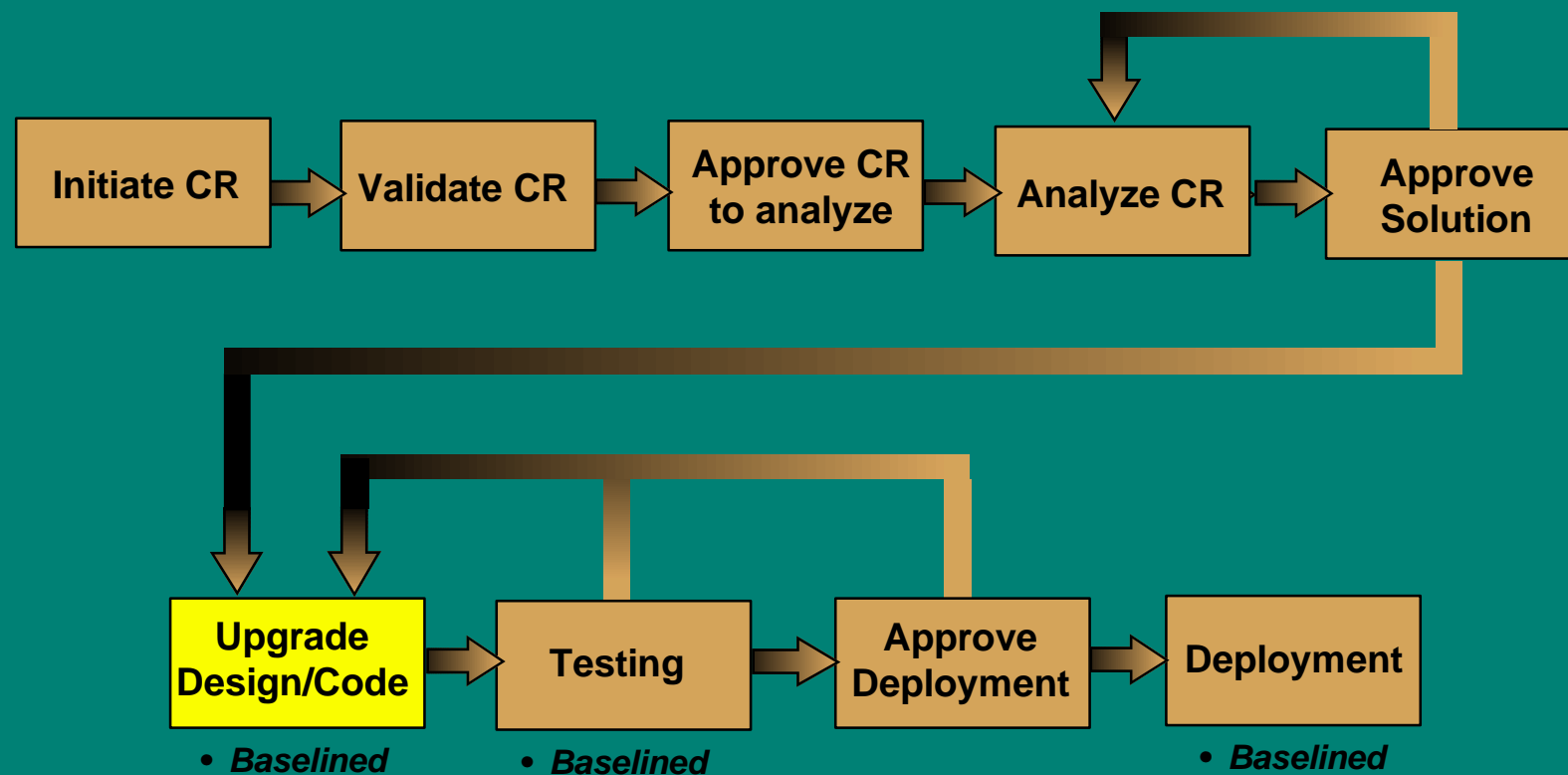
Approve Solution



Approve Solution

USER/TOOL	DESCRIPTION
Review Team (CFT, RM, CAB or SAB)	22 -- Approves solution and schedules implementation 23 -- Updates CR status to <i>“Solution Approved”</i>
Scopus/Oracle	24 -- Captures the CR status and implementation schedule 25 -- Notifies Review Team, Originator, Assignee, Developers

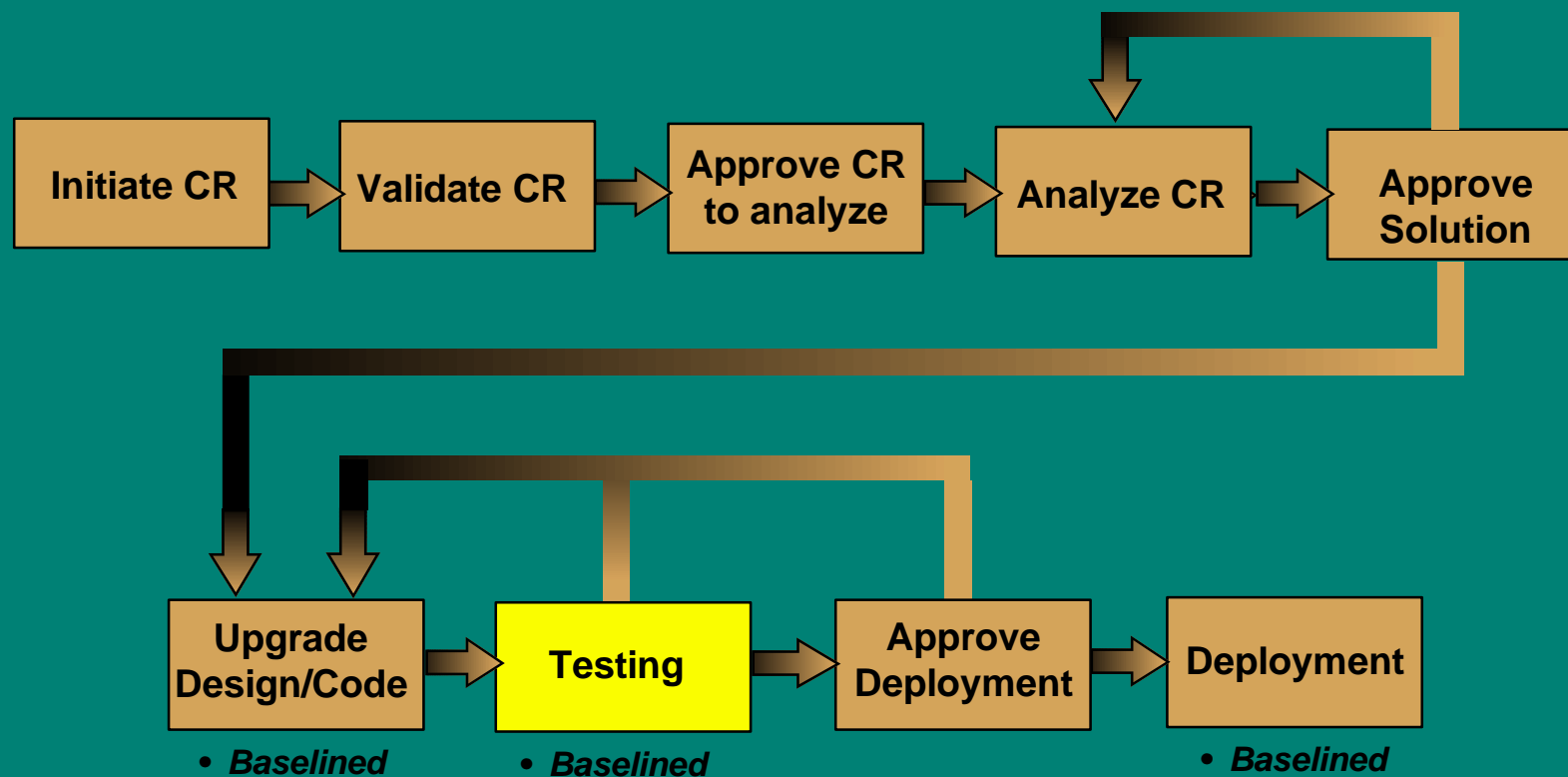
Upgrade Design/Code



Upgrade Design/Code

USER/TOOL	DESCRIPTION
CR Assignee	26 -- Identify required procurement items
Scopus/Oracle	27 -- Captures procurement items
Aide-de-Camp	28 -- Provides current baseline
Developer	29 -- Implements (Design/Code) and Documents Solution
System Architect	30 -- Captures new requirement / design
Developer	31 -- Conduct Unit testing 32 -- Updates the CR status to <i>“Unit Testing Complete”</i>
Tester	33 -- Write integration test scripts
Aide-de-Camp	34 -- Captures new version
Scopus/Oracle	35 -- Captures the CR status 36 -- Notifies Review Team, Originator & Assignee

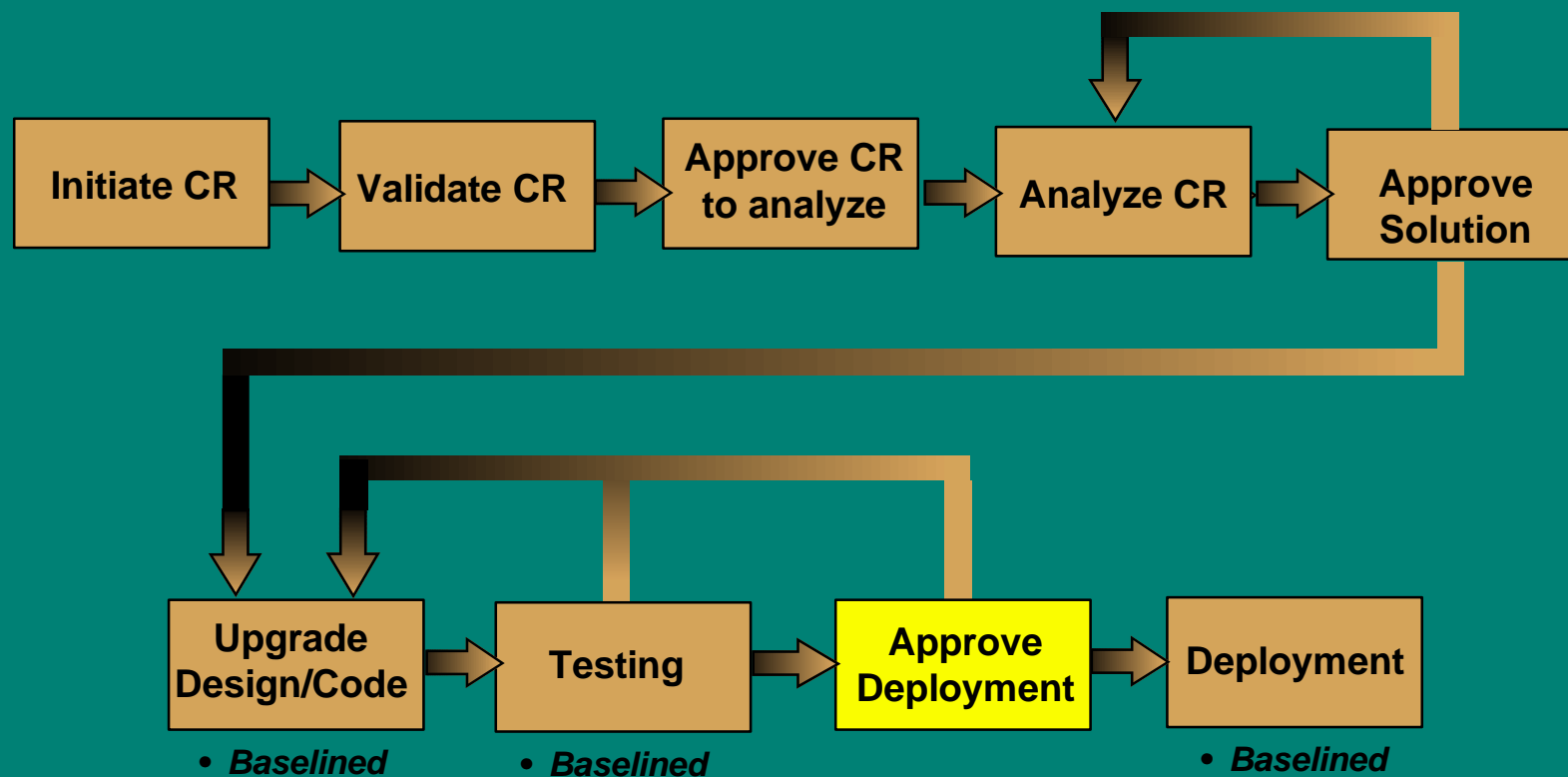
Testing



Testing

USER/TOOL	DESCRIPTION
Aide-de-Camp	37 -- Provides test suite 38 -- Provides baselines
Tester	39 -- Conducts integration testing 40 -- Conducts system testing 41 -- Conducts shadow testing 42 -- Updates test results 43 -- Updates CR status to <i>"Testing [IT, Sys, Shadow] Complete"</i>
Aide-de-Camp	44 -- Captures new baseline 45 -- Captures test results
Scopus/Oracle	46 -- Captures test status & CR status 47 -- Notifies Originator, Review Team & Assignee

Approve Deployment

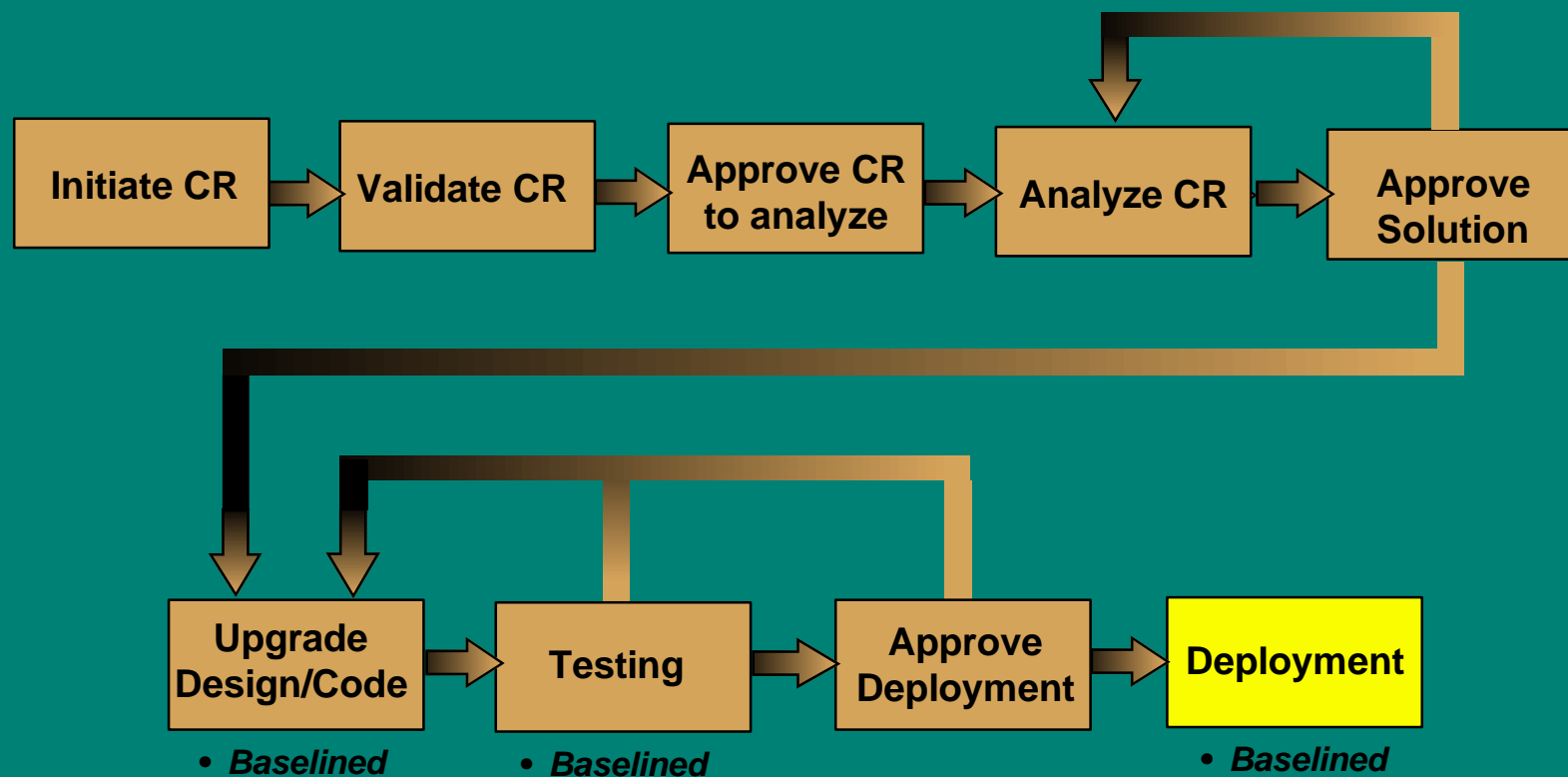




Approve Deployment

USER/TOOL	DESCRIPTION
Review Team (CFT, RM, CAB or SAB)	48 -- Reviews test results 49 -- Updates CR status to <i>“Release Approved”</i>
Aide-de-Camp	50 -- Captures new baseline
Scopus/Oracle	51 -- Captures the CR status 52 -- Notifies Review Team, Originator & Assignee

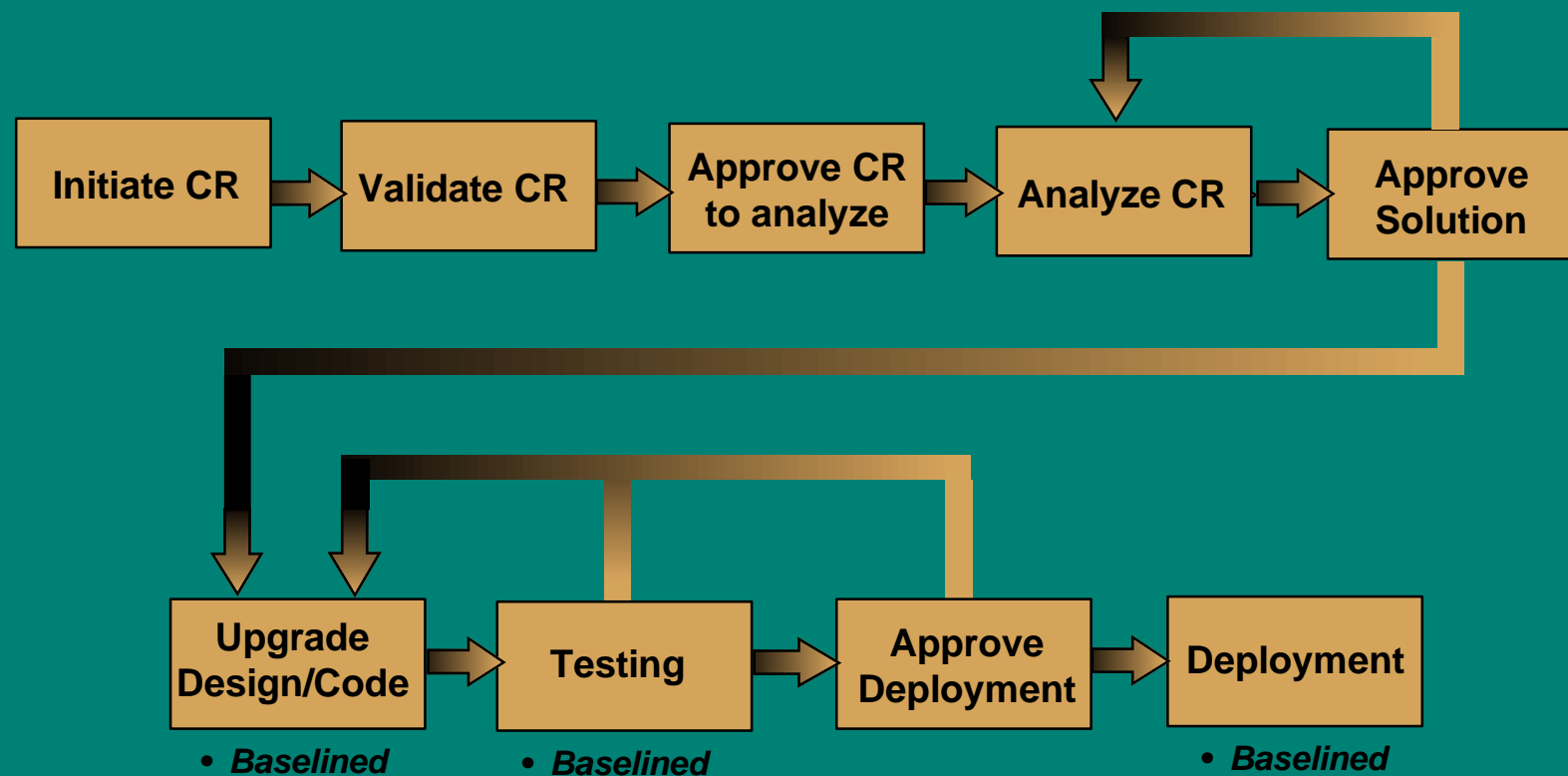
Deployment



Deployment

USER/TOOL	DESCRIPTION
Aide-de-Camp	53 -- Provides baseline and supports deployment
Customer	54 -- Installs change in operational environment 55 -- Updates CR status to <i>"Closed"</i>
Scopus/Oracle	56 -- Captures CR status 57 -- Notifies Originator, Review Team & Assignee
Customer	58 -- Backs out installation if necessary
Aide-de-Camp	59 -- Provides and reverts back to previous baseline
Scopus/Oracle	60 -- Notifies appropriate people

Complete Change Process





Presentation Goals

*To describe the
end-to-end change request process and
demonstrate MUGSy as an
integrated environment supporting this
process*



MUGSy Demo

❖ Question & Answers